PUT
PATCH
DELETE

BORA COMPLETAR NOSSO SERVIDOR?

HTTP - CRUD

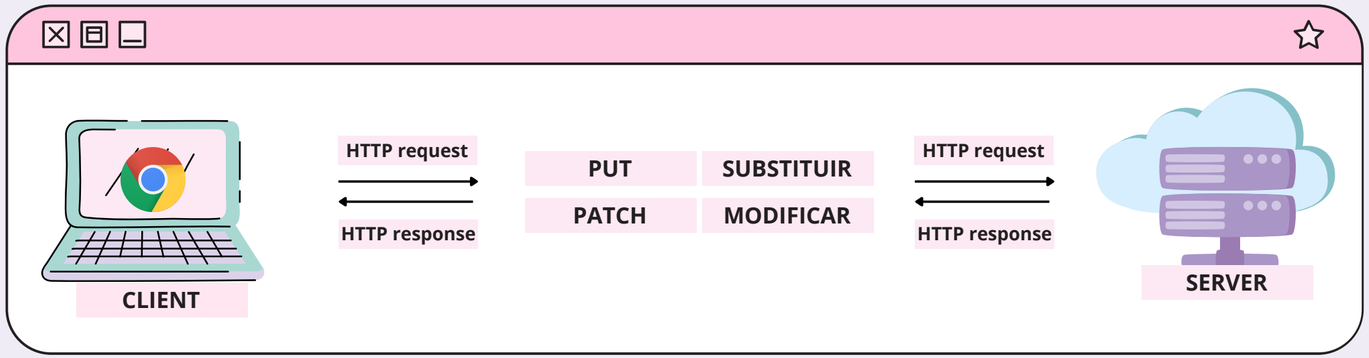
CRUD é a composição da primeira letra de **quatro** operações básicas de um banco de dados, e são o que a maioria das aplicações fazem.

C: Create (criar) - criar um novo registro
R: Read (ler) - exibir as informações de um registro
U: Update (atualizar) - atualizar os dados do registro
D: Delete (apagar) - apagar um registro

GET	READ
POST	CREATE
PATCH	UPDATE
PUT	UPDATE
DELETE	DELETE

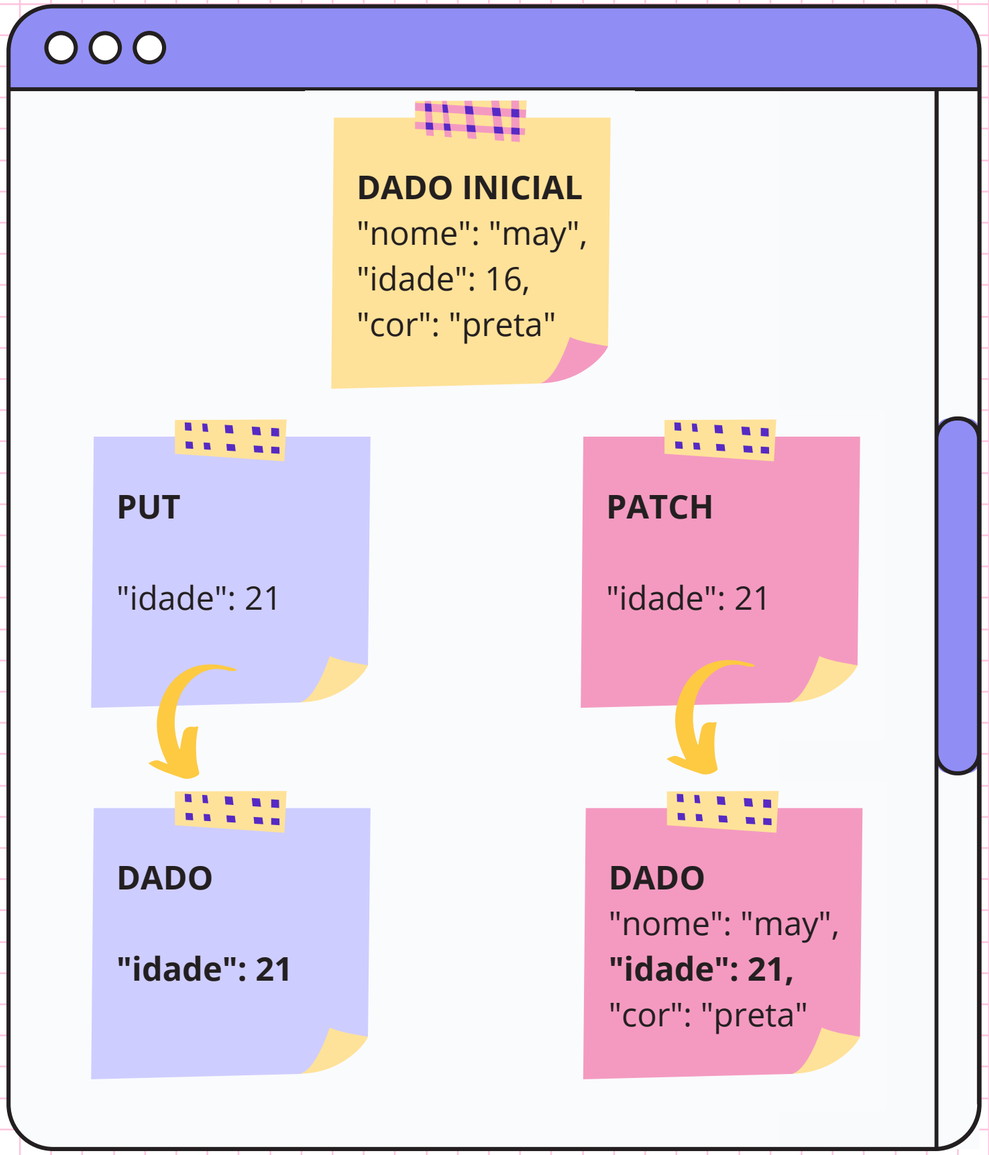
HTTP - PUT & PATCH

U: Update (atualizar) - atualiza os dados do registro.



É a mesma coisa, Prof?

NÃO! O PUT substitui todo o objeto que você deseja modificar, já o PATCH modifica somente uma propriedade dentro do seu objeto.

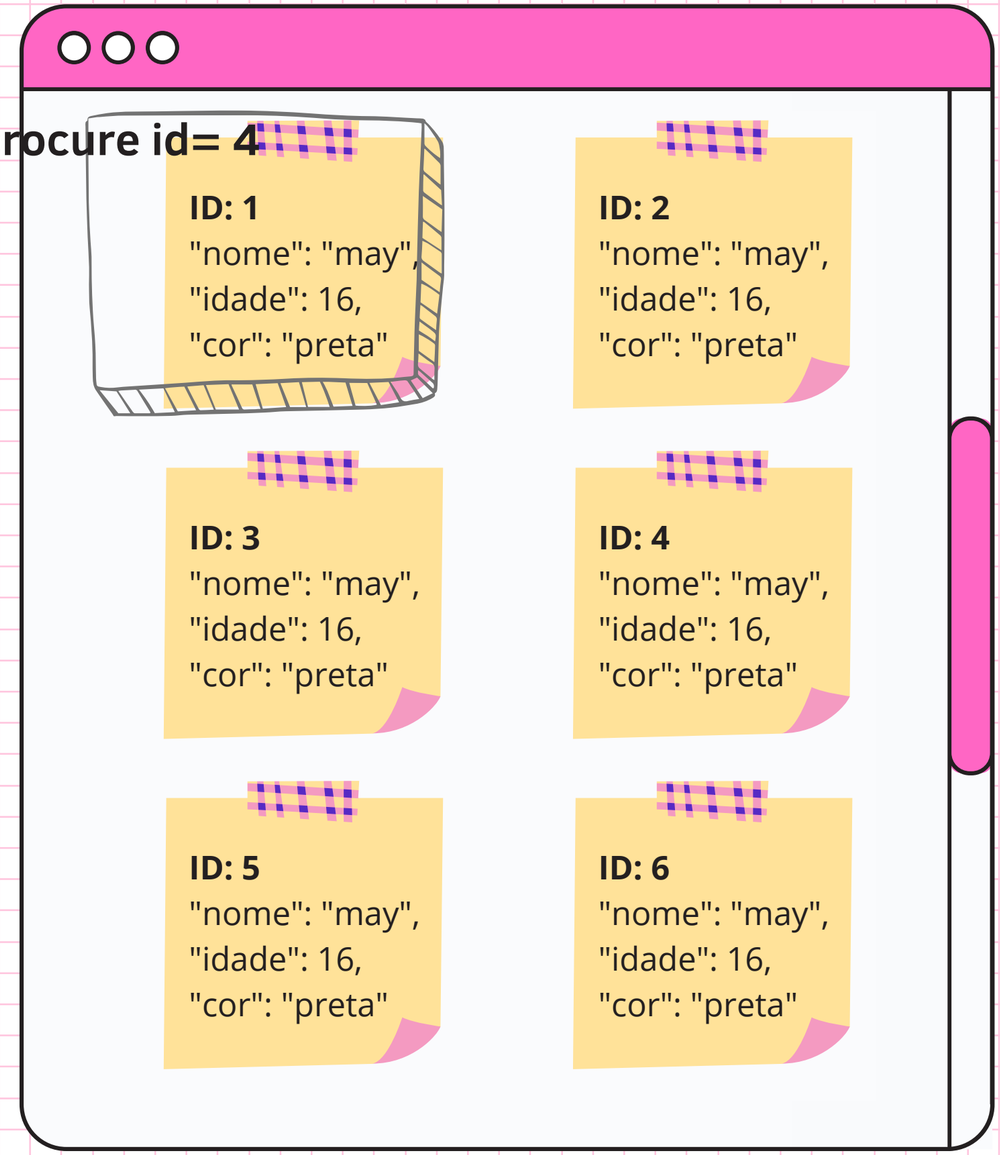


HTTP - PUT & PATCH

- Mas então por que ainda usamos o PUT?

Muitas vezes ainda usamos o PUT pela performance que ele tem quando relacionado a banco de dados. Substituir um dado inteiro é mais rápido do que somente uma propriedade dele.

Por exemplo, vamos simular a uma edição do campo de idade no dado de id=4

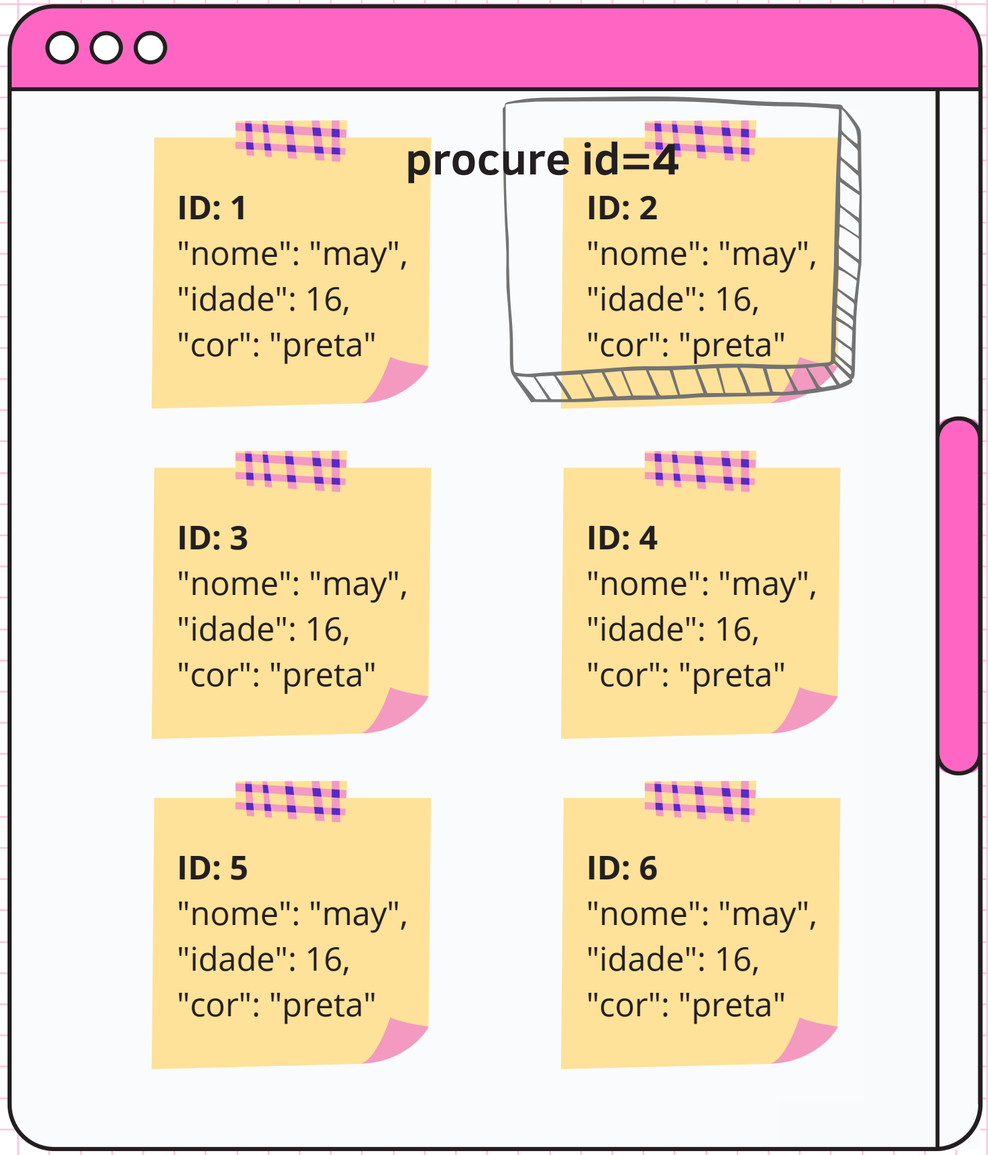


HTTP - PUT & PATCH

- Mas então por que ainda usamos o PUT?

Vamos simular a uma edição do campo de idade no dado de id=4

No banco de dados nosso programa tem que percorrer pela memória procurando pelo id que queremos;



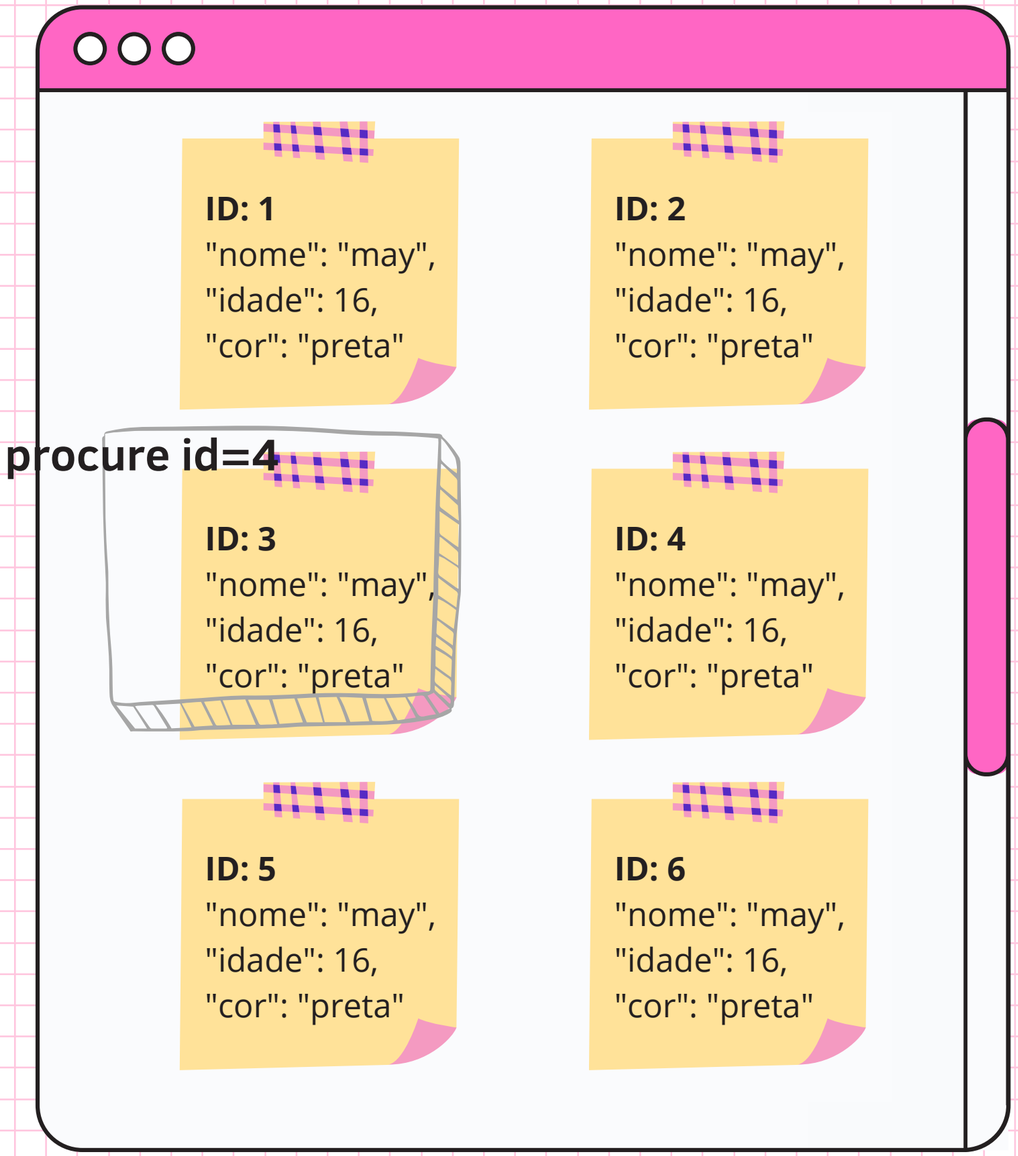
HTTP - PUT & PATCH

- Mas então por que ainda usamos o PUT?

Vamos simular a uma edição do campo de idade no dado de id=4

No banco de dados nosso programa tem que percorrer pela memória procurando pelo id que queremos;

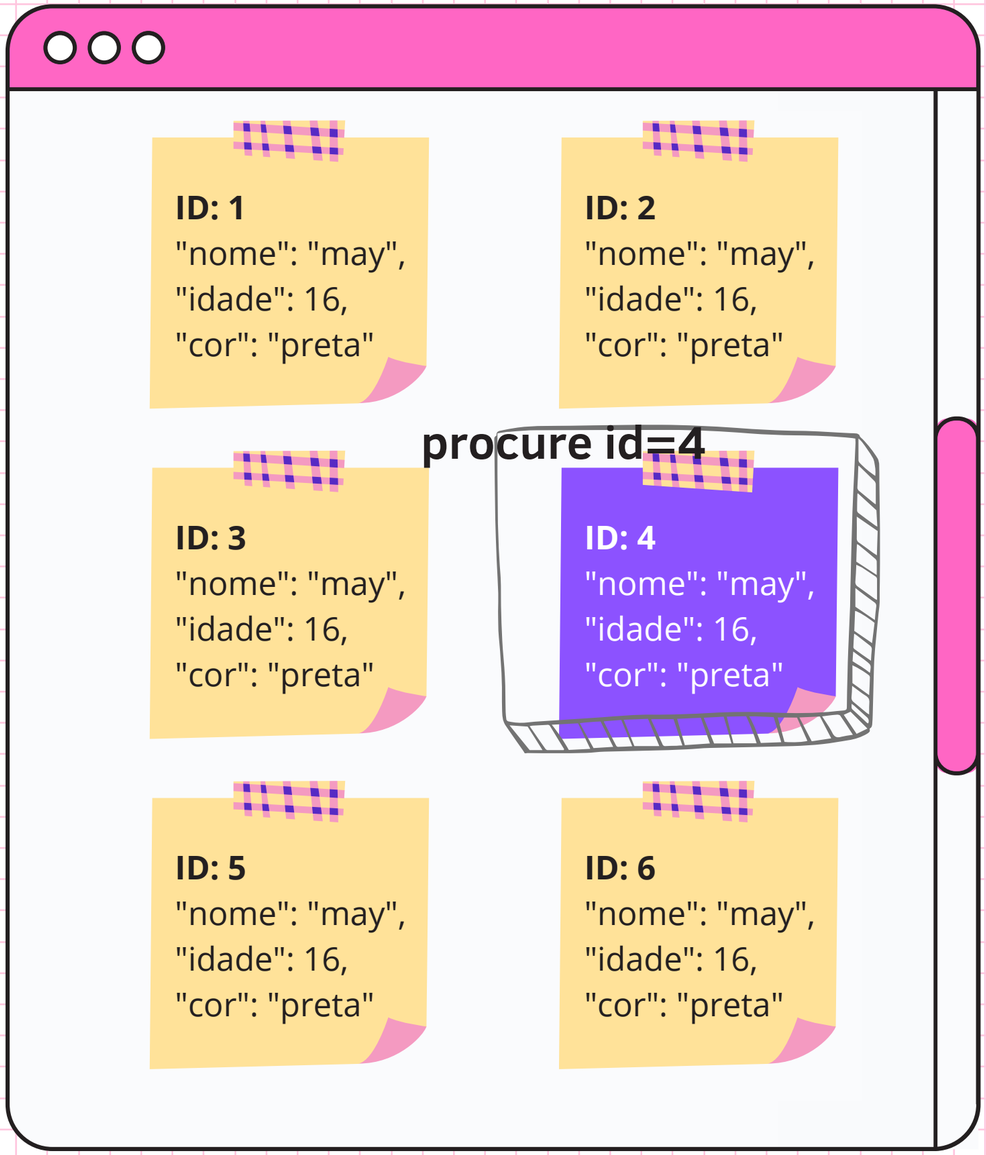
Ele procura somente pelo índice que indicamos



HTTP - PUT & PATCH

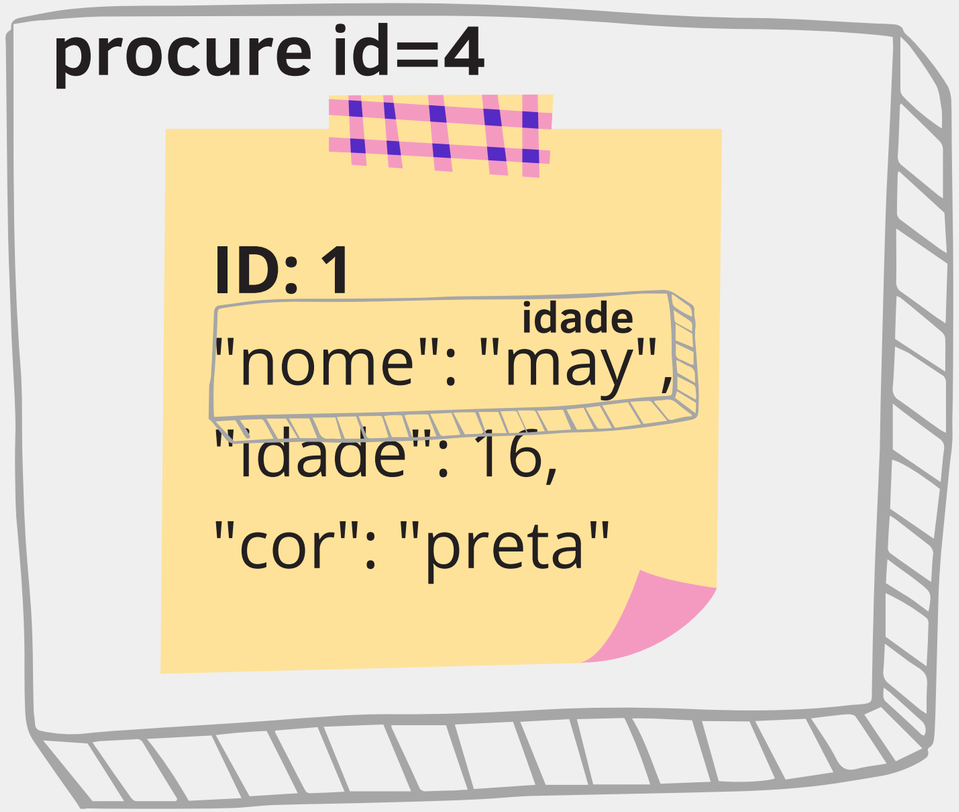
- Mas então por que ainda usamos o PUT?

Se escolhermos o método PUT, a procura pararia por aqui e o dado seria substituído por inteiro!



HTTP - PUT & PATCH

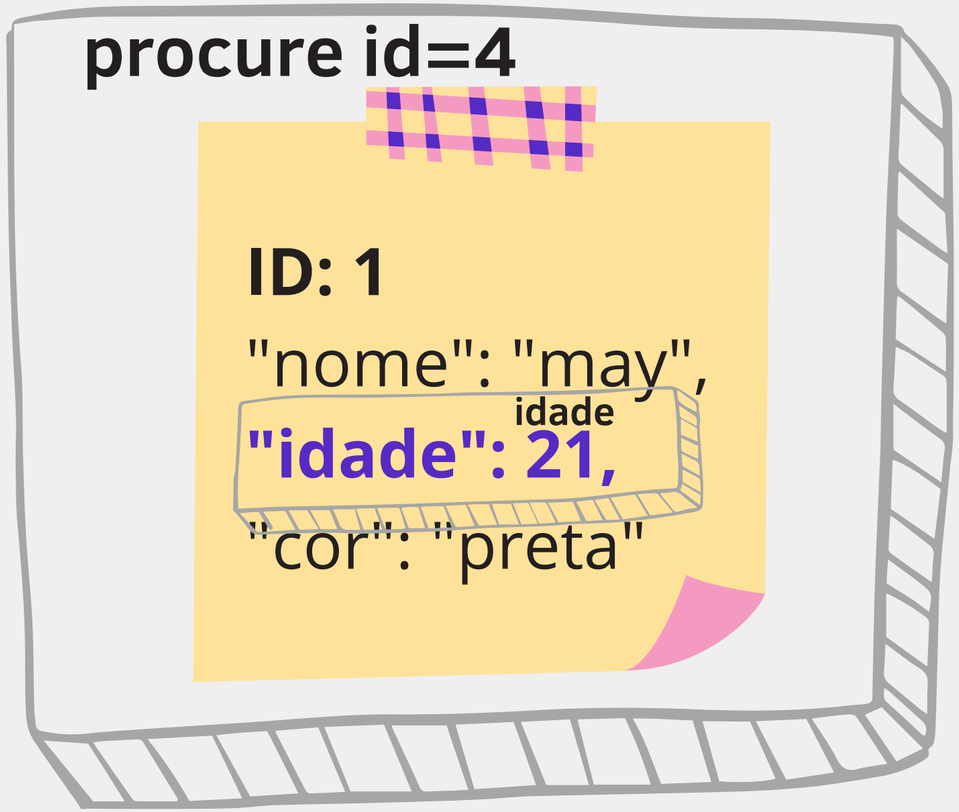
Em contrapartida, se escolhermos um método **PATCH**, a procura continuaria, só que agora é dentro do dado.



HTTP - PUT & PATCH

Quando encontrado a propriedade aí sim o dado seria modificado.

Tudo isso seriam frações de segundos para computador, mas se tivéssemos dezenas de milhares de dados sendo modificados o tempo todo, como uma rede social, por exemplo, isso poderia causar uma certa lentidão no banco de dados.



HTTP - DELETE

Usamos o método **DELETE** para **remover** um recurso ou uma coleção de recursos.

Quando em um formulário você clica no botão de “Excluir”, o evento que está sendo disparado passa pelos recursos do método DELETE.

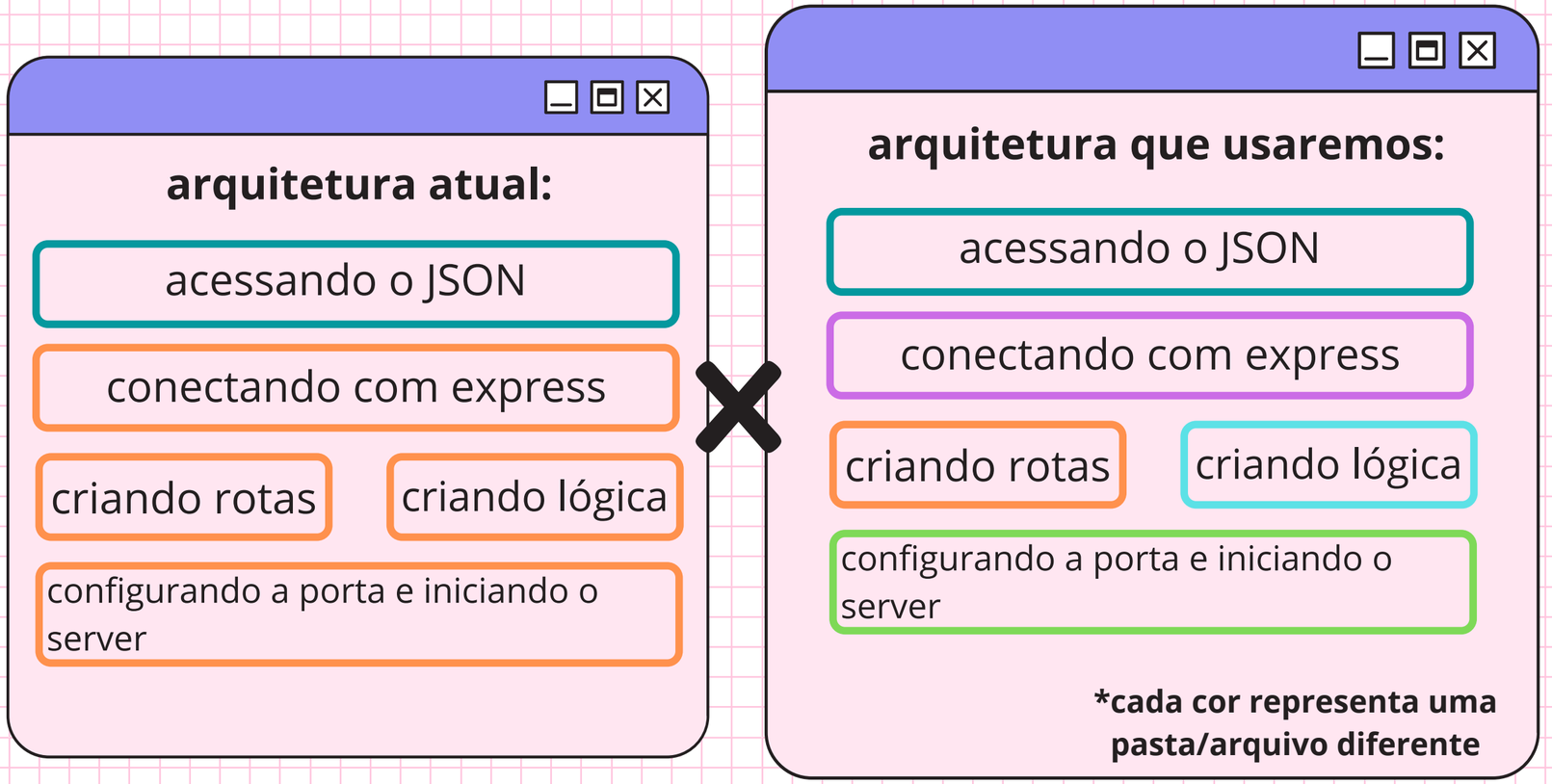
Esse foi fácil, né? hahahahah

ARQUITETURA MVC

MVC é um padrão de arquitetura de software, separando sua aplicação em 3 camadas. A camada de interação do usuário(view), a camada de manipulação dos dados(model) e a camada de controle(controller)

Já que estamos lidando com um projeto que tem somente back-end, não lidaremos com as views, porém lidamos com as rotas(routes).

O MVC nada mais é que uma forma de organizar o nosso código



ARQUITETURA MVC

server.js

configurando a porta e iniciando o servidor

app.js

indicação das rotas e requisições

filmesRoutes.js

rotas e verbos

pasta routes

filmesController.js

lógica

pasta controller

filmes.json

armazenamento

pasta model

arquitetura que usaremos:

acessando o JSON

conectando com express

criando rotas

criando lógica

configurando a porta e iniciando o server

***cada cor representa uma pasta/arquivo diferente**

PRA VER MELHOR

- nome-do-meu-server
  - .gitignore
  - package.json
  - package-lock.json
  - server
  - src
    - app.js
    - controller
      - nome-do-meu-controller.js
    - model
      - nome-do-meu-model.json
    - routes
      - nome-da-minha-route.js

TAREFINHA DA SEMANA

De acordo com a atividade da semana passada, você deverá escolher um dos models da nossa pastinha (filmes, pokemon ou todolist) e criar rotas de PUT, PATCH e DELETE.

GHLIBI FILMES

- Quero uma rota que altere todo os dados de um filme, com exceção do id;
- Devo conseguir alterar duração e/ou score do filme.
- Devo conseguir excluir filme por id;
- Devo conseguir excluir filme por diretor;

POKÉMON

- Quero uma rota que altere todos os dados de um pokemon, com exceção do id;
- Devo conseguir alterar stats de um pokemon.
- Devo conseguir excluir pokemon por id;
- Devo conseguir excluir pokemon por tipo;

TO DO LIST

- Quero uma rota que altere toda a tarefa, com exceção do id;
- Devo conseguir alterar nome da categoria;
- Devo conseguir excluir tarefas por id;
- Devo conseguir excluir todas as tarefas por categoria(id);